CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 1-11, 14, and 16-22.
- After this Amendment: Claims 1-11, 14, and 16-22.

Non-Elected, Canceled, or Withdrawn claims: None.

Amended claims: 2-6, and 11.

New claims: none.

Claims:

1. (Previously presented) One or more computer storage readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to:

access a configuration file written in a markup language and associated with an application, the configuration file having definitions of a plurality of configuration handlers for creating handler components based on a mapping table defined in the configuration file, at least one of the plurality of configuration handler definitions including a definition of a first configuration handler, the first configuration handler being configured to create a first handler component based on the mapping table and further comprising a second configuration handler nested within the first configuration handler, the second configuration handler being configured to create a second handler component based on a mapping definition in a configuration section within the first configuration handler, wherein the second configuration handler is user-defined to handle a customized handler data that is not supported by the first configuration handler, and wherein the second handler component created by the second configuration handler implements a known interface such that the data defined by the second configuration handler is properly processed by the application without alteration to the application;

create the plurality of handler components including the first handler component and the second handler component in accordance with the definitions;

inform one or more of the plurality of handler components of the presence of other handler components; and

make the plurality of handler components available to the application.

2. (Currently amended) One or more computer storage readable media as recited in claim 1, each of the plurality of configuration handler definitions being written in an extensible Markup Language (XML) format.

3. (Currently amended) One or more computer storage readable media as recited in claim 1, wherein to inform each of the plurality of handler components of the other handler components is to invoke a method exposed by one or more of the plurality of handler components.

4. (Currently amended) One or more computer storage readable media as recited in claim 3, wherein to invoke the method exposed by one or more of the plurality of handler components is further to include, as a parameter of the method, an identification of the plurality of handler components.

5. (Currently amended) One or more computer storage readable media as recited in claim 3, wherein the method comprises a WireUp method.

6. (Currently amended) One or more computer storage readable media as recited in claim 1, wherein the plurality of instructions, when executed by the one or more processors, further cause the one or more processors to implement nested configuration handlers defined in the configuration file.

7. (Previously presented) A method of using a configuration file to generate one or more handler components that are accessible to an application, the method comprising:

creating, in a first phase, a plurality of handler components defined in a configuration file, the creating comprising creating at least a first handler component based on a first configuration handler and a second handler component based on a second configuration handler, the second configuration handler being configured to nest within the first configuration handler, wherein the second handler component is user-defined to handle a customized handler data that is not supported by the first configuration handler, and wherein the second handler component implements a known interface such that the data defined by the second configuration handler is properly processed by the application without alteration to the application; and

notifying, in a second phase, one or more of the plurality of handler components of the presence of the other handler components.

8. (Previously presented) A method as recited in claim 7, the creating comprising:

obtaining, from the configuration file, definitions for each of the plurality of handler components;

identifying, from the configuration file, a configuration handler to be used to create one handler component of the plurality of handler components based on one of the definitions; and

while creating the one component, identifying, from the configuration file, a child configuration handler to be used to create another handler component to be used by the one handler component.

9. (Previously presented) A method as recited in claim 7, the notifying comprising:

invoking a method exposed by each of the one or more of the plurality of handler components.

10. (Previously presented) A method as recited in claim 9, the invoking comprising:

passing, as a parameter of the method, an identification of the plurality of handler components.

11. (Currently amended) One or more computer storage readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to access a configuration file and create a plurality of handler ~~component~~ components including at least a first and a second handler component ~~components~~ based on a first and a second configuration handler defined in the configuration file associated with an application, the first configuration handler comprising the second configuration handler and mapping the second configuration handler in a configuration section nested within the first configuration handler, wherein:

the second configuration handler is user-defined to handle a customized handler data that is not supported by the first configuration handler; ~~, wherein~~

the second handler component created by the second configuration handler implements a known interface such that the data defined by the second configuration handler is properly processed by the application without alteration to the application; and ~~, and~~

the first and the second handler components, upon creation, are notified of a presence of ~~notify~~ one or more of the plurality of handler components that have been previously created ~~of the presence of the other handler components~~.

12 (Cancelled).

13. (Cancelled)..

14. (Previously presented) A method comprising:

receiving a request to create a plurality of components from a configuration file associated with an application;

obtaining, from the configuration file, definitions for each of the plurality of handler components;

identifying, from the configuration file, a configuration handler to be used to create one component of the plurality of components based on one of the definitions;

while creating the one handler component, identifying, from the configuration handler, a child configuration handler to be used to create another handler component to be used by the one component, wherein the child configuration handler is nested within the configuration handler, wherein the child configuration handler is user-defined to handle a customized handler data that is not supported by the configuration handler, and wherein the other handler component created by the child configuration handler implements a known interface such that the data defined by the child configuration handler is properly processed by the application without alteration to the application;

notifying one or more of the plurality of handler components of the presence of the other handler components; and

making the plurality of handler components available to the application.

15. (Cancelled).

16. (Previously presented) A method as recited in claim 14, the identifying, from the configuration file, a child configuration handler comprising:

accessing a configuration section in the identified configuration handler, the configuration section mapping component identifiers to child configuration handlers; and

locating, from the mapping, the child configuration handler based on an identifier of the other component.

17. (Previously presented) A method as recited in claim 16, the identifier of the other component comprising an eXtensible Markup Language (XML) tag.

18. (Previously presented) A method as recited in claim 14, the definitions for each of the plurality of handler components being written in an eXtensible Markup Language (XML) format.

19. (Previously presented) A method as recited in claim 14, the identifying comprising:

identifying a tag associated with a definition of the one handler component;

accessing a mapping of tags to configuration handlers in the configuration file; and

identifying, using the mapping and based on the identified tag, the configuration handler to be used to create the one handler component.

20. (Previously presented) A method as recited in claim 19, the identifying, from the configuration file, a child configuration handler comprising:

accessing a configuration section in the identified configuration handler, the configuration section mapping component identifiers to child configuration handlers; and

locating, from the mapping, the child configuration handler based on an identifier of the other component.

21. (Previously presented) A system comprising:

a processor;

an application; and

a configuration system to access a configuration file associated with the application, the configuration file storing one or more extensible configuration handlers, the configuration system to create a plurality of handler components for the application in a two-phase process, the first phase including:

obtaining, from the configuration file, definitions for each of the plurality of handler components;

identifying, from the configuration file, a configuration handler to be used to create one handler component of the plurality of handler components based on one of the definitions; and

while creating the one handler component, identifying, from the one of the definitions, a child configuration handler to be used to create another handler component to be used by the one component, the child configuration handler being configured to nest in the definition of the one handler component, wherein the child configuration handler is user-defined to handle a customized handler data that is not supported by the configuration handler, and wherein the other handler component created by the child configuration handler implements a known interface such that the data defined by the child configuration handler is properly processed by the application without alteration to the application; and

the second phase including:

notifying one or more of the plurality of components of the presence of the other components in the plurality of components.

22. (Previously presented) A system as recited in claim 21, the notifying comprising:

invoking a method exposed by the one or more of the plurality of handler components, and

passing, as part of the invoking, the plurality of handler components as a parameter of the method.